Multimodal foundation world models for generalist embodied agents

Pietro Mazzaglia*
IDLab, Ghent University

Tim Verbelen
VERSES AI Research Lab

Bart Dhoedt
IDLab, Ghent University

Aaron Courville
Mila, University of Montreal

Sai Rajeswar
ServiceNow Research

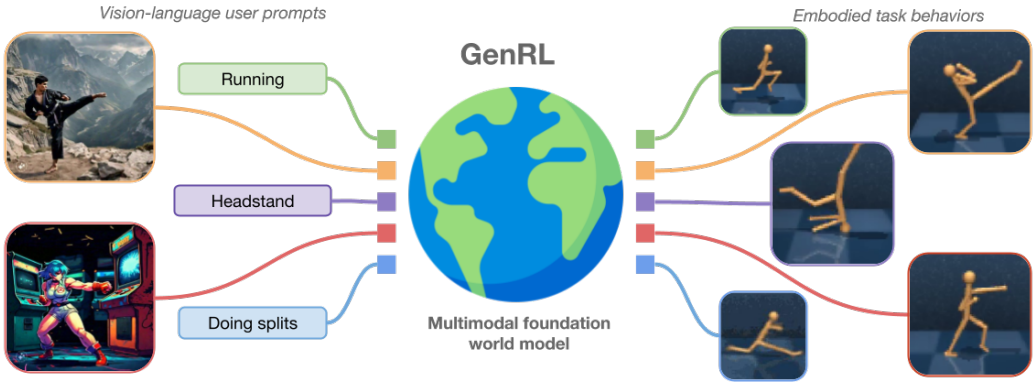


Figure 1: *Multimodal foundation world models* connect and align the video-language space of a foundation model with the latent space of a generative world model for reinforcement learning, requiring vision-only data. Our *GenRL* framework turns visual and/or language prompts into latent targets and learns to realize the corresponding behaviors by training in the world model's imagination.

Project website: `mazpie.github.io/genrl`

Abstract

Learning generalist embodied agents, able to solve multitudes of tasks in different domains is a long-standing problem. Reinforcement learning (RL) is hard to scale up as it requires a complex reward design for each task. In contrast, language can specify tasks in a more natural way. Current foundation vision-language models (VLMs) generally require fine-tuning or other adaptations to be functional, due to the significant domain gap. However, the lack of multimodal data in such domains represents an obstacle toward developing foundation models for embodied applications. In this work, we overcome these problems by presenting multimodal foundation world models, able to connect and align the representation of foundation VLMs with the latent space of generative world models for RL, without any language annotations. The resulting agent learning framework, GenRL, allows one to specify tasks through vision and/or language prompts, ground them in the embodied domain's dynamics, and learns the corresponding behaviors in imagination. As assessed through large-scale multi-task benchmarking, GenRL exhibits strong multi-task generalization performance in several locomotion and manipulation domains. Furthermore, by introducing a data-free RL strategy, it lays the groundwork for foundation model-based RL for generalist embodied agents.

*Work done while interning at Mila/ServiceNow Research. Email: `pietro.mazzaglia@ugent.be`

Preprint. Under review.

1 Introduction

Foundation models are large pre-trained models endowed with extensive knowledge of the world, which can be readily adapted for a given task [41]. These models have demonstrated extraordinary generalization capabilities in a wide range of vision [27, 42, 62] and language tasks [40, 18, 50, 11]. As we aim to extend this paradigm to embodied applications, where agents physically interact with objects and other agents in their environment, we require generalist agents that are capable of reasoning about these interactions and executing action sequences within these settings [57].

Reinforcement learning (RL) allows agents to learn complex behaviors from visual and/or proprioceptive inputs [17, 25, 26] by maximizing a specified reward function. Scaling up RL to multiple tasks and embodied environments remains challenging as designing reward functions is a complicated process, requiring expert knowledge and prone to errors which can lead to undesired behaviors [1]. Recent work has proposed the adoption of visual-language models (VLMs) to specify rewards for visual environments using language [4, 45], e.g. using the similarity score computed by CLIP [42] between an agent's input images and text prompts. However, these approaches require fine-tuning of the VLM [37] or adaptation of the visual domain [45], to work reliably.

In most RL settings, we lack multimodal data to train or fine-tune domain-specific foundation models, due to the costs of labelling agents' interactions and/or due to the intrinsic unsuitability of some embodied contexts to be converted into language. For instance, in robotics, it's non-trivial to convert a language description of a task to the agent's actions which are hardware-level controls, such as motor currents or joint torques. These difficulties make it hard to scale current techniques to large-scale generalization settings, leaving open the question:

How does one effectively leverage foundation models for generalization in embodied domains?

In this work, we present GenRL, a novel approach for training generalist agents from visual or language prompts, requiring no language annotations. GenRL learns multimodal foundation world models (MFWMs), where the joint embedding space of a foundation video-language model [54] is connected and aligned with the representation of a generative world model for RL [22], using unimodal vision-only data. The MFWM allows the specification of tasks by grounding language or visual prompts into the RL domain. Then, we introduce an RL objective for learning to accomplish the specified tasks in imagination [23], by matching the prompts in latent space.

Compared to previous work in world models and VLMs for RL, one emergent property of GenRL is the possibility to generalize to new tasks in a completely data-free manner. After training the MFWM, it possesses both strong priors over the dynamics of the environment, and large-scale multimodal knowledge. This combination enables the agent to interpret a large variety of task specifications and learn the corresponding behaviors. Thus, analogously to foundation models for vision and language, GenRL allows generalization to new tasks without additional data and lays the groundwork for foundation models in embodied RL domains [41]. [2]

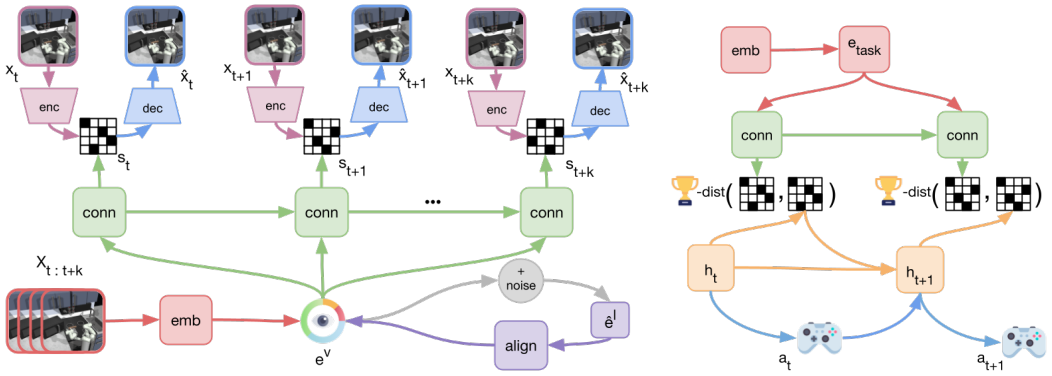
2 Preliminaries and background

Additional related works can be found in Appendix A.

Problem setting. The agent receives from the environment observations $x \in \mathcal{X}$ and interacts with it through actions $a \in \mathcal{A}$. The objective of the agent is to accomplish a certain task τ, which can be specified either in the observation space x_τ, e.g. through images or videos, or in language space y_τ, where $\mathcal{Y}$ represents the space of all possible sentences. Crucially, compared to a standard RL setting, we do not assume that a reward signal is available to solve the task. When a reward function exists, it is instead used to evaluate the agent's performance.

Generative world models for RL. In model-based RL, the optimization of the agent's actions is done efficiently, by rolling out and scoring imaginary trajectories using a (learned) model of the environment's dynamics. In recent years, this paradigm has grown successful thanks to the adoption of generative world models, which learn latent dynamics by self-predicting the agent's inputs [22]. World models have shown impressive performance in vision-based environments [23], improving our ability to solve complex and open-ended tasks [25]. Generative world models have

[2] The code, datasets and trained models will be made publicly available.



(a) Connecting and aligning (Section 3.2) (b) Learning task behavior (Section 3.3)

Figure 2: *Overview of GenRL.* The agent learns a multimodal foundation world model that connects and aligns (a) the representation of a foundation VLM with the latent states of a generative world model. Given a certain task prompt, (b) the model allows embedding the task and translating into targets in the latent dynamics space, which the agent can learn to achieve by using RL in imagination.

been successfully extended to many applications, such as exploration [48], skill learning [39], solving long-term memory tasks [47], and robotics [55].

Foundations models for RL. Large language models (LLMs) have been used for specifying behaviors using language [38, 28, 53], but this generally assumes the availability of a textual interface with the environment or that observations and/or actions can be translated to the language domain. The adoption of vision-language models (VLMs) reduces these assumptions, as it allows the evaluation of behaviors in the visual space. However. this approach has yet to show robust performance, as it generally requires fine-tuning of the VLM [4, 15], prompt hacking techniques [9] or visual modifications to the environment [4].

Vision-language generative modelling. Given the large success of image-language generative models [46], recent efforts in the community have focused on replicating and extending such success to the video domain, where the temporal dimension introduces new challenges, such as temporal consistency and increased computational costs [29, 3]. Video generative models are similar to world models for RL, with the difference that generation models outputs are typically not conditioned on actions, but rather conditioned on language[29] or on nothing at all (i.e. an unconditional model).

3 GenRL

3.1 World models for RL

GenRL learns a task-agnostic world model representation by modelling the sequential dynamics of the environment in a compact discrete latent space S [23, 25]. Latent states $s \in S$ are sampled from independent categorical distributions. The gradients for training the model are propagated through the sampling process with straight-through estimation [5].

The world model is made of the following components:

$$\begin{aligned} &\text{Encoder:} && q_\phi(s_t|x_t), && \text{Sequence model:} && h_t = f_\phi(s_{t-1}, a_{t-1}, h_{t-1}), \\ &\text{Decoder:} && p_\phi(x_t|s_t), && \text{Dynamics predictor:} && p_\phi(s_t|h_t), \end{aligned}$$

trained with the loss:

$$\mathcal{L}_\phi = \sum_t \underbrace{D_{\mathrm{KL}}\left[q_\phi(s_t|x_t) \| p_\phi(s_t|s_{t-1}, a_{t-1})\right]}_{\text{dyn loss}} - \underbrace{\mathbb{E}_{q_\phi(s_t|x_t)}[\log p_\phi(x_t|s_t)]}_{\text{recon loss}}, \tag{1}$$

where $p_\phi(s_t|s_{t-1}, a_{t-1})$ is a shorthand for $p_\phi(s_t|f_\phi(s_{t-1}, a_{t-1}, h_{t-1}))$. The sequence model is implemented as a linear GRU cell [8]. Differently from recurrent state space models (RSSM; [24]), for our framework, encoder and decoder models are not conditioned on the information present

in the sequence model. This ensures that the latent states only contain information about a single observation. We can then leverage the encoder as a probabilistic visual tokenizer, that is grounded in the target embodied environment.

3.2 Multimodal foundation world models

Multimodal VLMs are large pre-trained models that have the following components:

$$\text{Vision embedder:} \quad e^{(v)} = f_{\text{PT}}^{(v)}(x_{t:t+k}), \qquad \text{Language embedder:} \quad e^{(l)} = f_{\text{PT}}^{(l)}(y),$$

where $x_{t:t+k}$ is a sequence of visual observations and y is a text prompt. For video-language models, k is generally a constant number of frames (e.g. $k \in \{4, 8, 16\}$ frames). Image-language models are a special case where $k = 1$ as the vision embedder takes a single frame as an input. For our implementation, we adopt the InternVideo2 video-language model [54].

To connect the representation of the multimodal foundation VLM with the world model latent space, we instantiate two modules: a *latent connector* and a *representation aligner*:

$$\text{Connector:} \quad p_\psi(s_{t:t+k}|e), \qquad \mathcal{L}_{\text{conn}} = \sum_t D_{\text{KL}}\big[p_\psi(s_t|s_{t-1}, e) \| \text{sg}(q_\phi(s_t|x_t))\big],$$

$$\text{Aligner:} \quad e^{(v)} = f_\psi(e^{(l)}), \qquad \mathcal{L}_{\text{align}} = \|e^{(v)} - f_\psi(e^{(l)})\|_2^2,$$

where $\text{sg}(\cdot)$ indicates to stop gradients propagating.

The connector learns to predict the latent states of the world model from embeddings in the VLM's representation space. The connector's objective consist of minimizing the KL divergence between its predictions and the world model's encoder distribution. While more expressive architectures, such as transformers [52] or state-space models [20] could be adopted, we opt for a simpler GRU-based architecture for video modelling. This way, we keep the method simple and the architecture of the connector is symmetric with respect to the world model's components.

Multimodal VLMs trained with contrastive learning exhibit a multimodality gap [33], where the spherical embeddings of different modalities are not aligned. The role of the aligner is to reduce this multimodality gap, by projecting text embeddings into their corresponding visual embeddings. Given a dataset of vision-language data, this projective function can be learned.

In embodied domains vision-language data is typically unavailable. Instead, we leverage the idea that language embeddings can be treated as a 'corrupted' version of their vision counterparts [61, 63]. This allows us to approximate the language embedding with the corresponding vision embedding: $e^{(l)} \approx e^{(v)} + \epsilon$.

Previous methods inject the noise into the embeddings when training the new module [61, 63], in our case the connector. However, we opt for training a separate aligner network. This allows us to train a noise-free connector, which has two main advantages: (i) it yields higher prediction accuracy for visual embedding inputs while maintaining a similar alignment for language embedding inputs; and (ii) it is more flexible; it's easier to re-train/adapt for different noise levels, as it only requires re-training the aligner module, and its use can be avoided if unnecessary.

3.3 Learning specified task behaviors in imagination

World models can be used to imagine trajectories in latent space, using the dynamics model. This allows us to train behavior policies in a model-based RL fashion [23]. Given a task specified through a visual or language prompt, our MFWM can generate the corresponding latent states by turning the embedder's output, e_{task}, into sequences of latent states $s_{t:t+k}$ (examples are shown in Figure 1). The objective of the policy model π_θ is then to match the goals specified by the user by performing trajectory matching.

The trajectory matching problem can be solved as a divergence minimization problem [13], between the distribution of the states visited by the policy π_θ and the trajectory generated using the aligner-connector networks from the user-specified prompt:

$$\theta = \arg\min_\theta \mathbb{E}_{a_t \sim \pi_\theta(s_t)} \Big[\sum_t \gamma^t \texttt{distance}\big(p_\phi(s_{t+1}|s_t, a_t) \| p_\psi(s_{t+1}|e_{\text{task}})\big)\Big], \quad \text{with} \quad e_{\text{task}} = f_{\text{PT}}(\cdot). \tag{2}$$

The KL divergence is a natural candidate for the distance function [13]. However, in practice, we found that using the cosine distance between linear projections of the latent states notably speeds up learning and enhances stability. We can then turn the objective in Eq. 2 into a reward for RL:

$$r_{\text{GenRL}} = \cos\big(g_\phi(s^{\text{dyn}}_{t+1}), g_\phi(s^{\text{task}}_{t+1})\big), \quad \text{with} \quad s^{\text{dyn}}_{t+1} \sim p_\phi(s_{t+1}|s_t, a_t), s^{\text{task}}_{t+1} \sim p_\psi(s_{t+1}|e_{\text{task}}), \quad (3)$$

where g_ϕ represents the first linear layer of the world model's decoder. We train an actor-critic model to maximize this reward and achieve the tasks specified by the user [25]. Additional implementation details are provided in Appendix B.

Temporal alignment. One issue with trajectory matching is that it assumes that the distribution of states visited by the agent starts from the same state as the target distribution. However, the initial state generated by the connector may differ from the initial state where the policy is currently in. For example, consider the Stickman agent on the right side of Figure 1. If the agent is lying on the ground and tasked to run, the number of steps to get up and reach running states may surpass the temporal span recognized by the VLM (e.g. typically 4, 8, or 16 frames), causing disalignment in the reward.

To address this initial condition alignment issue, we propose a *best matching trajectory* technique, inspired by best path decoding in speech recognition [19]. Our technique involves two steps:

1. We compare the first b states of the target trajectory with b states obtained from the trajectories imagined by the agent by sliding along the time axis. This allows one to find at which timestep t_a the trajectories are best aligned (the comparison provides the highest reward).
2. We align the temporal sequences in the two possible contexts: (a) if a state from the agent sequence comes before t_a, the reward uses the target sequence's initial state; and (b) if the state comes k steps after t_a, it's compared to the s_{t+k} state from the target sequence.

In all experiments, we fix $b = 8$ (number of frames of the VLM we use [54]), which we found to strike a good compromise between comparing only the initial state ($b = 1$) and performing no alignment ($b =$ imagination horizon). An ablation study can be found in Appendix E.

4 Experiments

Overall, we employ a set of 4 locomotion environments (Walker, Cheetah, Quadruped, and a newly introduced Stickman environment) [51] and one manipulation environment (Kitchen) [21], for a total of 35 tasks where the agent is trained without rewards, only from text prompts. This represents the first large-scale study of multi-task generalization from language in RL. Details about datasets, tasks, and prompts used can be found in the Appendix C.

4.1 Offline RL

In offline RL, the objective of the agent is to learn to extract a certain task behavior from a given fixed dataset [32]. The performance of the agent generally depends on its ability to 'retrieve' the correct behaviors in the dataset and interpolate among them. Popular techniques for offline RL include off-policy RL methods, such as TD3 [17], advantage-weighted behavior cloning, such as IQL [30], and behavior-regularized approaches, such as CQL [31] or TD3+BC [16].

We aim to assess the multi-task capabilities of different approaches for designing rewards using VLMs. We collected large datasets for each of the domains evaluated, containing a mix of structured data (i.e. the replay buffer of agent learning to perform some tasks) and unstructured data (i.e. exploration data collected using [48]). We have removed the explicit reward information about the task and replaced it with a short task description, in language form.

We compare GenRL to two main categories of approaches:

- *Image-language rewards*: following [45], the cosine similarity between the embedding for the language prompt and the embedding for the agent's visual observation is used as a reward. For the VLM, we adopt the SigLIP-B [60] model as it's reported to have superior performance than the original CLIP [42].
- *Video-language rewards*: similar to the image-language rewards, with the difference that the vision embedding is computed from a video of the history of the last k frames, as done in [15]. For the VLM, we use the InternVideo2 model [54], the same used for GenRL.

Table 1: *Behavior extraction.* Offline RL from language prompts on tasks that are included in the agent's training dataset. Scores are episodic rewards averaged over 5 seeds (± standard error) rescaled using min-max scaling with $(\min = \text{random policy}, \max = \text{expert policy})$.

	Image-language reward			Video-language reward			Ours
	IQL	TD3+BC	TD3	IQL	TD3+BC	TD3	GenRL
walker stand	0.68 ± 0.03	0.97 ± 0.01	0.92 ± 0.06	0.72 ± 0.05	0.59 ± 0.05	**1.0 ± 0.0**	**1.01 ± 0.0**
walker run	0.24 ± 0.03	0.27 ± 0.02	0.11 ± 0.03	0.26 ± 0.02	0.25 ± 0.02	0.35 ± 0.01	**0.76 ± 0.01**
walker walk	0.37 ± 0.04	0.25 ± 0.03	0.15 ± 0.0	0.43 ± 0.04	0.39 ± 0.01	0.86 ± 0.04	**0.95 ± 0.03**
cheetah run	0.37 ± 0.07	-0.01 ± 0.0	-0.01 ± 0.0	0.15 ± 0.03	-0.01 ± 0.0	0.34 ± 0.01	**0.74 ± 0.01**
quadruped stand	0.32 ± 0.06	0.4 ± 0.06	0.62 ± 0.04	0.39 ± 0.04	0.41 ± 0.13	0.73 ± 0.06	**0.98 ± 0.0**
quadruped run	0.3 ± 0.03	0.35 ± 0.01	0.25 ± 0.02	0.36 ± 0.04	0.35 ± 0.04	0.23 ± 0.02	**0.61 ± 0.04**
quadruped walk	0.16 ± 0.02	0.18 ± 0.02	0.37 ± 0.02	0.19 ± 0.04	0.16 ± 0.04	0.37 ± 0.02	**0.73 ± 0.06**
stickman run	0.2 ± 0.02	0.25 ± 0.02	0.03 ± 0.0	0.24 ± 0.02	0.18 ± 0.03	0.21 ± 0.0	**0.35 ± 0.02**
stickman walk	0.43 ± 0.07	0.52 ± 0.05	0.18 ± 0.01	0.47 ± 0.02	0.45 ± 0.08	0.42 ± 0.03	**0.78 ± 0.02**
stickman stand	0.52 ± 0.05	**0.68 ± 0.04**	0.56 ± 0.04	0.56 ± 0.06	0.47 ± 0.02	0.61 ± 0.03	0.49 ± 0.04
kitchen microwave	0.14 ± 0.13	0.0 ± 0.0	0.0 ± 0.0	0.02 ± 0.02	0.0 ± 0.0	0.46 ± 0.18	**0.67 ± 0.27**
kitchen light	0.18 ± 0.12	**1.0 ± 0.0**	0.66 ± 0.07	0.04 ± 0.02	0.0 ± 0.0	0.0 ± 0.0	0.43 ± 0.19
kitchen burner	0.14 ± 0.07	0.43 ± 0.14	0.02 ± 0.02	0.02 ± 0.02	0.0 ± 0.0	0.28 ± 0.08	**0.67 ± 0.14**
kitchen slide	0.16 ± 0.1	0.0 ± 0.0	0.0 ± 0.0	0.06 ± 0.04	0.0 ± 0.0	**1.0 ± 0.0**	**1.0 ± 0.0**
overall	0.30 ± 0.04	0.38 ± 0.02	0.28 ± 0.02	0.28 ± 0.02	0.23 ± 0.02	0.49 ± 0.04	**0.73 ± 0.05**

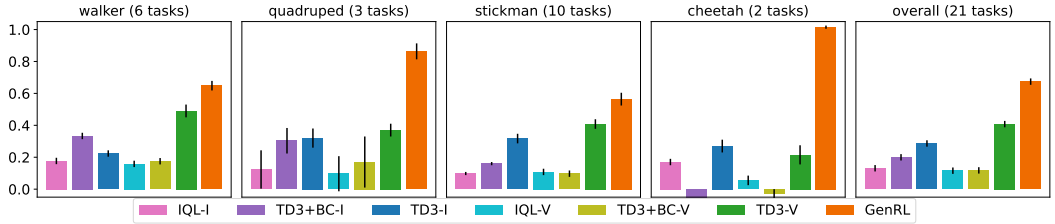


Figure 3: *Multi-task generalization.* Offline RL from language prompts on tasks that are not deliberately included in the training dataset. Performance averaged over 5 seeds and standard error was reported with black lines. Detailed results per task in Appendix E.

We test both approaches with a variety of offline RL methods, including IQL, TD3+BC, and TD3. All methods are trained for 500k gradient steps, and evaluated on 20 episodes. Other training details are reported in Appendix D.

Behavior extraction. We want to verify whether the methods can retrieve the tasks behaviors that are certainly present in the dataset. We present results in Table 1, with episodic rewards rescaled so that 0 represents the performance of a random agent, while 1 represents the performance of an expert agent.

GenRL excels in overall performance across all domains, particularly in dynamic tasks like walking and running in the quadruped and cheetah domains. In contrast, in static tasks such as 'stickman stand' and kitchen tasks, other methods occasionally outperform GenRL. This can be explained by the fact that the target sequences that GenRL infers from the prompt are often slightly in motion, even in static cases. To address this, we could set the target sequence length to 1 for static prompts, but we opted to maintain the method's simplicity and generality, acknowledging this as a minor limitation.

As expected, video-language rewards tend to perform better than image-language rewards for dynamic tasks. For video-based rewards, the less conservative approach, TD3, performs better than all other baselines in most tasks, similarly to what is shown in [58]. We instead observe an opposite trend for image-language rewards, where more conservative approaches, such as IQL and TD3+BC, tend to perform better. We believe this is because when the 'task target' is static, imitating segments of trajectories from the dataset proves beneficial.

Multi-task generalization. To assess multi-task generalization, we defined a set of tasks not included in the training data. Although we don't anticipate agents matching the performance of expert models, higher scores in this benchmark help gauge the generalization abilities of different methods. We

averaged the performance across various tasks for each domain and summarized the findings in Figure 3, with detailed task results in Appendix E.

Overall, we observe a similar trend as for the behavior extraction results. GenRL significantly outperforms the other approaches, especially in the quadruped and cheetah domains, where the performance is close to the specialized agents' performance. Both for image-language (-I in the Figure) and video-language (-V in the Figure) more conservative approaches, such as IQL and TD3+BC tend to perform worse. This could be associated with the fact that imitating segments of trajectories is less likely to lead to high-rewarding trajectories, as the tasks are not present in the training data.

4.2 Data-free RL

In the previous section, we evaluated several approaches for designing reward using foundation VLMs. Clearly, model-free RL approaches require continuous access to a dataset, to train the actor-critic and generalize across new tasks. Model-based RL can learn the actor-critic in imagination. However, in previous work [25, 23], imagining sequences for learning behaviors first requires processing actual data sequences. The data is used to initialize the dynamics model, and obtain latent states that represent the starting states to rollout the policy in imagination. Furthermore, in order to learn new tasks, reward-labelled data is necessary to learn a reward model, which provides rewards to the agent during the task learning process.

Foundation models [41] are generally trained on enormous datasets in order to generalize to new tasks. The datasets used for the model pretraining are not necessary for the downstream applications, and sometimes these datasets are not even publicly available [40, 18]. In this section, we aim to establish a new paradigm for foundation models in RL, which follows the same principle of foundation models for vision and language. We call this paradigm *data-free RL* and we define it as the ability to generalize to new tasks, after pre-training, using no additional data.

GenRL enables data-free RL thanks to two main reasons: the agent learns a task-agnostic MFWM on a large varied dataset during pre-training, and the MFWM enables the possibility of specifying tasks directly in latent space, without requiring any data. Thus, in order to learn behaviors in imagination, the agent can: `(i)` sample random latent states in the world model's representation, `(ii)` rollout sequences in imagination, following the policy, and `(iii)` compute rewards, using the targets obtained by processing the given prompts with the connector-aligner networks.

Initial states distribution. Uniform sampling from the latent space of the world model often results in meaningless latent states. Additionally, the sequential dynamics model of the MFWM, using a GRU, requires some 'warmup' steps to discern dynamic environmental attributes, such as velocities.

To address these issues we perform two operations. First, we combine uniformly sampled states from the discrete latent spaces with states generated by randomly sampling the connector model, as sequences generated by the connector tend to have a more coherent structure than random uniform samples. Second, we perform a rollout of five steps using a mix of actions from the trained policy and random actions. This leads to a varied distribution of states, containing dynamic information, which we use as the initial states for the learning in imagination process.

Performance of data-free RL. Our results, detailed in Figure 4, compare data-free RL to traditional offline RL with GenRL, as discussed in Section 4.1. We ablate the choice of using random samples from the connector-aligner to improve randomly sampled initial states. While data-free RL generally

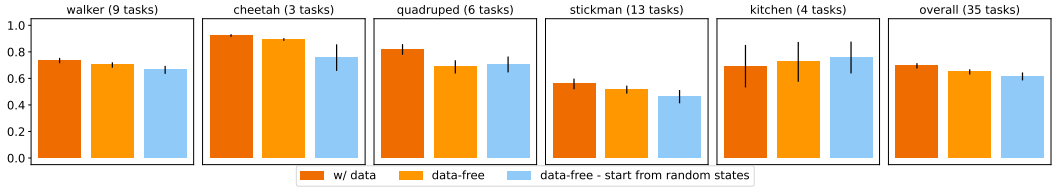


Figure 4: *Data-free RL.* Generalization to new tasks, learning behaviors in imagination without relying on any data for initializing the sequences to learn in imagination. Performance is averaged over 5 seeds and standard error is reported with black lines. Detailed results per task in Appendix E.

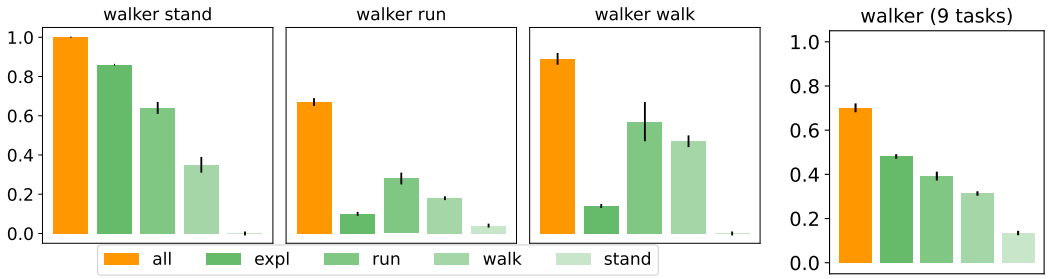


Figure 5: *Training data distribution*. Analysing the impact of the training data distribution on the generalization performance of GenRL. Performance is obtained by training behaviors in data-free mode, after training the MFWM on different subsets of the training dataset. Performance averaged over 3 seeds (black lines indicate standard error). Full results in Appendix E.

shows a slight decrease in overall performance, the differences are minimal across most domains, and it even outperforms in the kitchen domain. The use of states from the connector model enhances average scores and reduces variance, especially noticeable in the cheetah domain.

By employing data-free learning, after pre-training, agents can master new tasks without data, often converging within only 30 minutes of GPU training. As we scale up foundation models for behavior learning, the ability to learn data-free will become crucial. Although very large datasets will be employed to train new foundation models, GenRL adapts well without direct access to original data, offering flexibility where data may be proprietary, licensed or unavailable.

4.3 Training data distribution

As demonstrated in Sections 4.1 and 4.2, after training on a large dataset, a GenRL agent can adapt to multiple new tasks without additional data. The nature of the training data, detailed in Appendix C, combines exploration and task-specific data. To identify critical data types for GenRL, we trained different MFWMs on various dataset subsets. Then, we employ data-free RL to train task behaviors, with analyses over subsets of the walker dataset provided in Figure 5, where 'all' reports the data-free performance when training on the full dataset.

The results confirm that a diverse data distribution is crucial for task success, with the best performance achieved by using the complete dataset, followed by the varied exploration data. Task-specific data effectiveness depends on task complexity, for instance, 'run' data proves more generalizable than 'walk' or 'stand' data across tasks. Crucially, 'stand' data, which shows minimal variation, limits learning for a general agent but can still manage simpler tasks like 'lying down' and 'sitting on knees' as detailed in Appendix E.

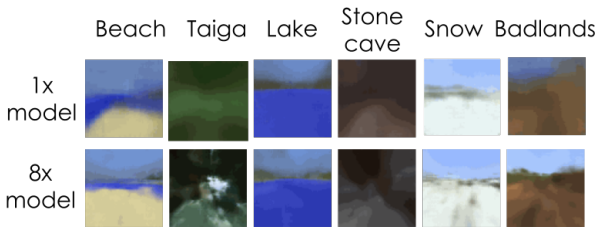
Moving forward with training foundation models in RL, it will be essential to develop methods that extract multiple behaviors from unstructured data and accurately handle complex behaviors from large datasets. Thus, the ability of GenRL to primarily leverage unstructured data is a significant advantage for scalability.

5 Additional Analysis

A framework for behavior generation. A common challenge with using LLMs and VLMs involves the need for prompt tuning to achieve specific tasks. As GenRL relies on a foundation VLM, similar to previous approaches [4, 45] it is not immune from this issue. However, GenRL uniquely allows for the visualization of targets obtained from specific prompts. By decoding the latent targets, using the MFWM decoder, we can visualize the interpreted prompt before training the corresponding behavior. This enables a much more explainable framework, which allows fast iteration for prompt tuning, compared to previous (model-free) approaches which often require training the agent to identify which behaviors are rewarded given a certain prompt.

Video grounding. In Section 4, we have focussed our evaluation on language-driven RL, using text prompts to specify tasks. While language strongly simplifies specification of the task, in some cases providing visual examples of the task might be easier. Similarly as for language prompts, GenRL

(a) Video grounding examples



(b) Decoded language prompts in Minecraft

allows translating vision prompts (short videos) into behaviors. As this is a less common evaluation setting, we limit our assessment to qualitative results.

In Figure 6a, we present a set of video grounding examples, obtained by inferring the latent targets corresponding to the vision prompts (right image) and then using the decoder model to decode images (left image). We observe that the agent is able to translate short human action videos[3] into the same actions but for the Stickman embodiment. By applying this approach, it would be possible to learn behaviors from a single video.

Scaling to complex observations. Generalist embodied agents should be able to scale to open-ended learning settings. Using GenRL, we explored this by training an agent in the Minecraft environment using a small dataset collected by a DreamerV3 agent [25]. The primary challenge we found was the model's difficulty in reconstructing complex observations in this open-ended environment.

Reconstructing complex observations is a common issue with world models [10]. To overcome this limitation, while keeping the method unaltered, we attempted to scale up the number of parameters of MFWM. Qualitative reconstruction results are presented in Figure 6b. We observe that the agent is able to identify different biomes from language, even with the smaller size of the model. However, the reconstructions are significantly blurrier compared to the other environments we analyzed (e.g. Figure 6a). When using a larger model, the reconstructions gain some details but the results still highlight the difficulty of the model in providing accurate targets from prompts.

While this might not be an issue for simple high-level tasks, e.g. 'navigate to a beach', unclear targets might make it difficult to perform more precise actions, e.g. 'attack a zombie'. Future research should aim to address this issue, for instance, by improving our simple GRU-based architecture, leveraging transformers or diffusion models to improve the quality of the representation [29, 3].

6 Discussion

We introduced GenRL, a world-model based approach for grounding vision-language prompts into embodied domains and learning the corresponding behaviors in imagination. The multimodal foundation world models of GenRL can be trained using unimodal data, overcoming the lack of multimodal data in embodied RL domains. The data-free RL property of GenRL lays the groundwork for foundation models in RL that can generalize to new tasks without additional data.

Limitations. Despite its strengths, GenRL presents some limitations, largely due to inherent weaknesses in its components. From the VLMs, GenRL inherits the issue related to the multimodality gap [33, 61] and the reliance on prompt tuning. We proposed a connection-alignment mechanism to mitigate the former. For the latter, we presented an explainable framework, which facilitates prompt tuning by allowing decoding of the latent targets corresponding to the prompts. From the world model, GenRL inherits a dependency on reconstructions, which offers advantages such as explainability but also drawbacks, such as failure modes with complex observations.

Future work. As we strive to develop foundation models for generalist embodied agents, our framework opens up numerous research opportunities. One such possibility is to learn multiple behaviors and have another module, e.g. an LLM, compose them to solve long-horizon tasks. Another promising area of research is investigating the temporal flexibility of the GenRL framework. We witnessed that for static tasks, greater temporal awareness could enhance performance. This concept could also apply to actions that extend beyond the time comprehension of the VLM. Developing general solutions to these challenges could lead to significant advancements in the framework.

[3] Short videos are generated using `meta.ai`.

Acknowledgments

Pietro Mazzaglia is funded by a Ph.D. grant of the Flanders Research Foundation (FWO). This research was supported by a Mitacs Accelerate Grant.

References

[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety, 2016.

[2] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos, 2022.

[3] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, G. Liu, A. Raj, Y. Li, M. Rubinstein, T. Michaeli, O. Wang, D. Sun, T. Dekel, and I. Mosseri. Lumiere: A space-time diffusion model for video generation, 2024.

[4] K. Baumli, S. Baveja, F. Behbahani, H. Chan, G. Comanici, S. Flennerhag, M. Gazeau, K. Holsheimer, D. Horgan, M. Laskin, et al. Vision-language models as a source of rewards. *arXiv preprint arXiv:2312.09187*, 2023.

[5] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[6] J. Bruce, M. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. Behbahani, S. Chan, N. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktäschel. Genie: Generative interactive environments, 2024.

[7] E. Cetin, A. Tirinzoni, M. Pirotta, A. Lazaric, Y. Ollivier, and A. Touati. Simple ingredients for offline reinforcement learning, 2024.

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[9] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran. Can foundation models perform zero-shot task specification for robot manipulation?, 2022.

[10] F. Deng, I. Jang, and S. Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 4956–4975. PMLR, 2022.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[12] Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[13] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth. Model-based imitation learning by probabilistic trajectory matching. In *2013 IEEE international conference on robotics and automation*, pages 1922–1927. IEEE, 2013.

[14] A. Escontrela, A. Adeniji, W. Yan, A. Jain, X. B. Peng, K. Goldberg, Y. Lee, D. Hafner, and P. Abbeel. Video prediction models as rewards for reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35:18343–18362, 2022.

[16] S. Fujimoto and S. S. Gu. A minimalist approach to offline reinforcement learning, 2021.

[17] S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods, 2018.

[18] Gemini Team et al. Gemini: A family of highly capable multimodal models, 2024.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[20] A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[21] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning, 2019.

[22] D. Ha and J. Schmidhuber. World models. 2018.

[23] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination, 2020.

[24] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels, 2019.

[25] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[26] N. Hansen, H. Su, and X. Wang. Td-mpc2: Scalable, robust world models for continuous control, 2024.

[27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

[28] M. Klissarov, P. D'Oro, S. Sodhani, R. Raileanu, P.-L. Bacon, P. Vincent, A. Zhang, and M. Henaff. Motif: Intrinsic motivation from artificial intelligence feedback, 2023.

[29] D. Kondratyuk, L. Yu, X. Gu, J. Lezama, J. Huang, R. Hornung, H. Adam, H. Akbari, Y. Alon, V. Birodkar, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[30] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning, 2021.

[31] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning, 2020.

[32] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[33] W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning, 2022.

[34] S. Lifshitz, K. Paster, H. Chan, J. Ba, and S. McIlraith. Steve-1: A generative model for text-to-behavior in minecraft, 2024.

[35] J. Lin, Y. Du, O. Watkins, D. Hafner, P. Abbeel, D. Klein, and A. Dragan. Learning to model the world with language, 2023.

[36] H. Liu, W. Yan, M. Zaharia, and P. Abbeel. World model on million-length video and language with blockwise ringattention, 2024.

[37] E. S. Lubana, J. Brehmer, P. de Haan, and T. Cohen. Fomo rewards: Can we cast foundation models as reward functions?, 2023.

[38] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. Eureka: Human-level reward design via coding large language models, 2024.

[39] P. Mazzaglia, T. Verbelen, B. Dhoedt, A. Lacoste, and S. Rajeswar. Choreographer: Learning and adapting skills in imagination. In *International Conference on Learning Representations*, 2023.

[40] OpenAI et al. Gpt-4 technical report, 2024.

[41] R. Bommasani et al. On the opportunities and risks of foundation models, 2022.

[42] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.

[43] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[44] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022.

[45] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

[46] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[47] M. R. Samsami, A. Zholus, J. Rajendran, and S. Chandar. Mastering memory tasks with world models, 2024.

[48] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models, 2020.

[49] D. Tarasov, V. Kurenkov, A. Nikulin, and S. Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning, 2023.

[50] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.

[51] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[53] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023.

[54] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, J. Xu, Z. Wang, Y. Shi, T. Jiang, S. Li, H. Zhang, Y. Huang, Y. Qiao, Y. Wang, and L. Wang. Internvideo2: Scaling video foundation models for multimodal video understanding, 2024.

[55] P. Wu, A. Escontrela, D. Hafner, K. Goldberg, and P. Abbeel. Daydreamer: World models for physical robot learning, 2022.

[56] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, L. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators, 2024.

[57] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans. Foundation models for decision making: Problems, methods, and opportunities, 2023.

[58] D. Yarats, D. Brandfonbrener, H. Liu, M. Laskin, P. Abbeel, A. Lazaric, and L. Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning, 2022.

[59] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021.

[60] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training, 2023.

[61] Y. Zhang, E. Sui, and S. Yeung-Levy. Connect, collapse, corrupt: Learning cross-modal tasks with uni-modal data, 2024.

[62] L. Zhao, N. B. Gundavarapu, L. Yuan, H. Zhou, S. Yan, J. J. Sun, L. Friedman, R. Qian, T. Weyand, Y. Zhao, R. Hornung, F. Schroff, M.-H. Yang, D. A. Ross, H. Wang, H. Adam, M. Sirotenko, T. Liu, and B. Gong. Videoprism: A foundational visual encoder for video understanding, 2024.

[63] Y. Zhou, R. Zhang, C. Chen, C. Li, C. Tensmeyer, T. Yu, J. Gu, J. Xu, and T. Sun. Lafite: Towards language-free training for text-to-image generation, 2022.

A Extended Related Work

[Linked to Section 2]

World models. Recent research has focused on the question of how to learn world models from large-scale video datasets [36, 56]. In [6], they leverage a latent action representation, but their work is mostly focussed on 2D platform videogames or simple robotic actions. In [14], they use frame-by-frame video prediction as a way to provide rewards for RL. DynaLang [35] studies the incorporation of language prediction as part of the world model, to train multimodal world models also from datasets without actions or rewards. The representation in DynaLang is shared in the world model between vision and language, while for GenRL, the world model representation is trained on vision-only data and connected-aligned to the multimodal foundation representation.

Foundation models for actions. Few cases of foundation models for embodied domains have been developed. Notable mentions are GATO [44], a large-scale behavior cloning agent, trained on 604 tasks. VPT [2] a large-scale model trained on Minecraft data, using human-expert labeled trajectories. The model learns strong behavioral priors by behavior cloning which can be fine-tuned using RL. STEVE-1 [34] connects VPT's behavioral prior with the MineCLIP model representation [15], using the unCLIP approach [43]. RT-X [12] are large-scale trasnformer models trained on expert robotics dataset, sharing a common action space (end-effector pose) across different embodiments.

B Implementation details

[Linked to Section 3]

Actor-critic. Rewards can be maximized over time in imagination in a RL fashion, using actor-critic models of the form:

$$\text{Actor:} \quad \pi_\theta(a_t|s_t), \qquad \text{Critic:} \quad v_\theta(R_t^\lambda|s_t), \quad \text{where} \quad R_t^\lambda = r_t + \gamma[(1-\lambda v_{t+1}) + \lambda R_{t+1}^\lambda]$$

For the actor-critic, we follow the implementation advances proposed in DreamerV3 [25] (version 1 of the paper, dated January 2023), such as using a two-hot distribution for learning the critic network and scaling returns in the actor loss.

When computing the reward r_{GenRL}, we use the mode of the distribution for the target $s_{t+1}^{\text{task}} \sim p_\psi(s_{t+1}|e_{\text{task}})$ to improve stability.

Hyperparameters. For the hyperparameters, we follow DreamerV3 [25] (version 1 of the paper, dated January 2023). Differences from the default hyperparameters or model size choices are illustrated in Table 2. For instance, a main difference is that we use difference batch sizes/lengths for training the MFWM and the actor-critic as these two stages are now independent from each other.

The connector network uses the same hyperparameters and architecture as the sequential dynamics of the world model. The aligner network employs a small U-Net, with a bottleneck that is half the size of the embedding representation.

Name	**Value**
Multimodal Foundation World Model	
Batch size	48
Sequence length	48
GRU recurrent units	1024
CNN multiplier	48
Dense hidden units	1024
MLP layers	4
Actor-Critic	
Batch size	32
Sequence length	32

Table 2: World model and actor-critic hyperparameters.

C Tasks

[Linked to Section 4]

We present the list of tasks employed, along with the language prompts used for specifying the task, in Table 3. We introduce a new embodied environment, the 'Stickman', which serves as a humanoid robot (compared to the one present in the dm_control suite) that is simpler to control, thanks to a reduced number of joints. Its addition allows studying behaviors that involve upper body movements, rather than focusing on lower body motions. For the newly introduced tasks, the goal can be easily inferred by reading the task's name or its prompt. For the 'flipping' tasks, we consider flips both in forward direction and backward direction, as the VLM struggles to distinguish directions.

The prompts we use have been fine-tuned for the InternVideo2 model [54]. However, we found that they mostly improved performance for the SigLIP model too [60]. One common observation is that these models are

Table 3: Task and prompt used for each task

Task	Prompt	Specialized agent score	Random agent score
quadruped run	spider running fast	930	10
quadruped walk	spider walking fast	960	10
quadruped stand	spider standing	990	15
quadruped jump	spider jumping	875	15
quadruped two legs	on two legs	875	14
quadruped lie down	lying down	965	750
cheetah run	running like a quadruped	890	9
cheetah standing	standing like a human	930	5
cheetah lying down	lying down	920	430
stickman walk	robot walk fast clean	960	35
stickman run	robot run fast clean	830	25
stickman stand	standing	970	70
stickman flipping	doing flips	790	45
stickman one foot	stand on one foot	865	20
stickman high kick	stand up and kick	920	55
stickman lying down	lying down horizontally	965	380
stickman sit knees	praying	966	40
stickman lunge pose	lunge pose	950	100
stickman headstand	headstand	955	180
stickman boxing	punch	920	80
stickman hands up	standing with the hands up	830	5
walker walk	walk fast clean	960	45
walker run	run fast clean	770	30
walker stand	standing up straight	970	150
walker flipping	doing backflips	720	20
walker one foot	stand on one foot	955	20
walker high kick	stand up and kick	960	25
walker lying down	lying down horizontally	975	170
walker sit knees	praying	945	100
walker lunge pose	lunge pose	945	150
kitchen microwave	opening the microwave fully open	1	0
kitchen light	activate the light	1	0
kitchen burner	the burner becomes red	1	0
kitchen slide	slide cabinet above the knobs	1	0

generally biased towards human actions. Thus, specifying the embodiment in the prompt is sometimes helpful, e.g. ‘spider running fast’ or ‘running like a quadruped’. Another observation is that for some behaviors the agent can produce very different styles, e.g. the agent can be walking in a slow or fast way, or in a more or less composed manner. Specifying words like ‘fast’ or ‘clean’ helps clarifying what kind of behavior is expected.

D Experiments settings

[Linked to Section 4]

Baselines. In order to implement performant offline RL baselines we adopt the findings of [49] and [7], adopting larger deeper networks and layer normalization.

Inputs are 64x64x3 RGB images. We use a frame stack of 3. The encoder architecture is adapted from the DrQ-v2 encoder [59]. We did find augmentations on the images, e.g. random shifts, to hurt performance.

Offline RL. For each task, training model-free agents (IQL, TD3, TD3+BC) requires re-training the full agent (visual encoder, actor, critic) on the entire dataset, from scratch, while training model-based agents (GenRL) requires training the model once for each domain and then training an actor-critic for each task. Moreover, for training the actor-critic in GenRL, we only use 50k gradient steps, as the policy converges significantly faster than for the other methods.

Compute resources. We use a cluster of V100 with 16GB of VRAM for all our experiments. To enable efficient training, image and video-CLIP embeddings are computed in advance and stored with the datasets. Training the MFWM for 500k gradient steps takes $\sim$ 5 days. After pre-training the MFWM, training the actor-critic for a prompt for 50k gradient steps takes less than 5 hours. In data-free mode, it takes less than 3 hours. In both cases, convergence normally arrives after 10k gradient steps, but we keep training. Model-free baselines take around 7 hours to train for 500k gradient steps. normally arrive

Datasets composition. We present the datasets' composition in Table 4.

Table 4: Datasets composition.

Domain	$\sim$ **num of observations**	**Subset**	**Subcount**
walker	2.5M	walker run	500k
		walker walk	500k
		walker stand	500k
		walker expl	1M
cheetah	1.8M	cheetah run	1M
		cheetah expl	820k
quadruped	2.5M	quadruped expl	1M
		quadruped run	500k
		quadruped stand	500k
		quadruped walk	500k
kitchen	3.6M	kitchen slide	700k
		kitchen light	700k
		kitchen bottom burner	700k
		kitchen microwave	700k
		kitchen expl	800k
stickman	2.5M	stickman stand	500k
		stickman walk	500k
		stickman expl	1M
		stickman run	500k
minecraft	4M	-	-

E Additional results

[Linked to Section 4]

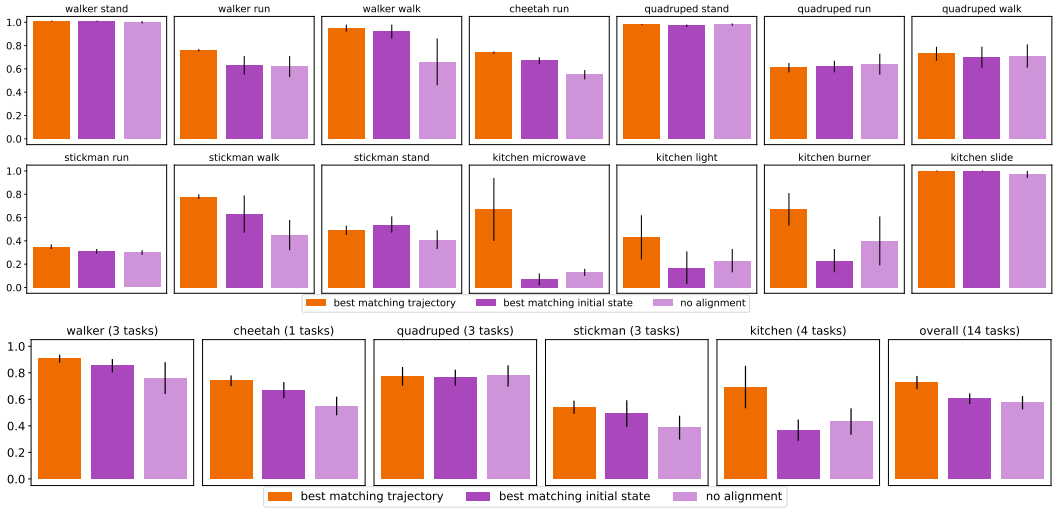


Figure 7: *Temporal alignment ablation.* We analyze the impact of temporal alignment in our proposed RL objective for matching sequential targets. Results averaged over 3 seeds.

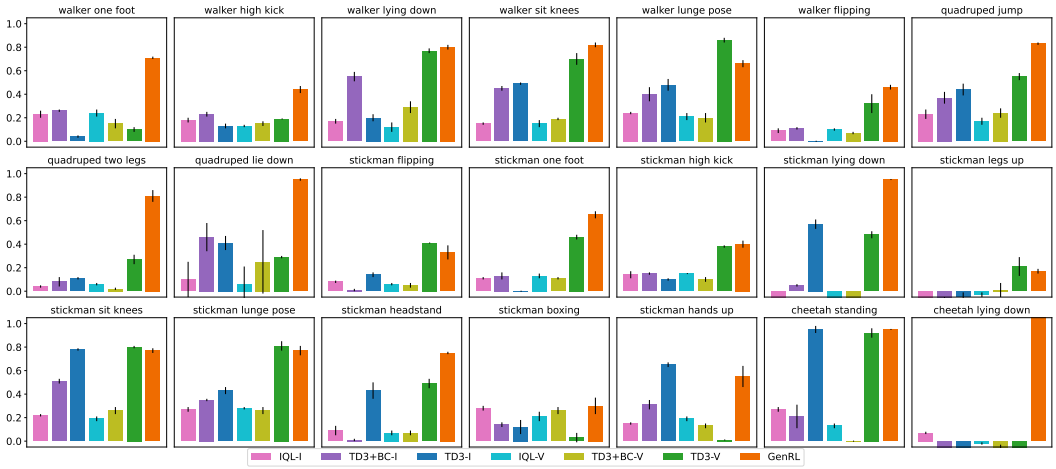


Figure 8: *Multi-task generalization detailed results.* Results averaged over 5 seeds.

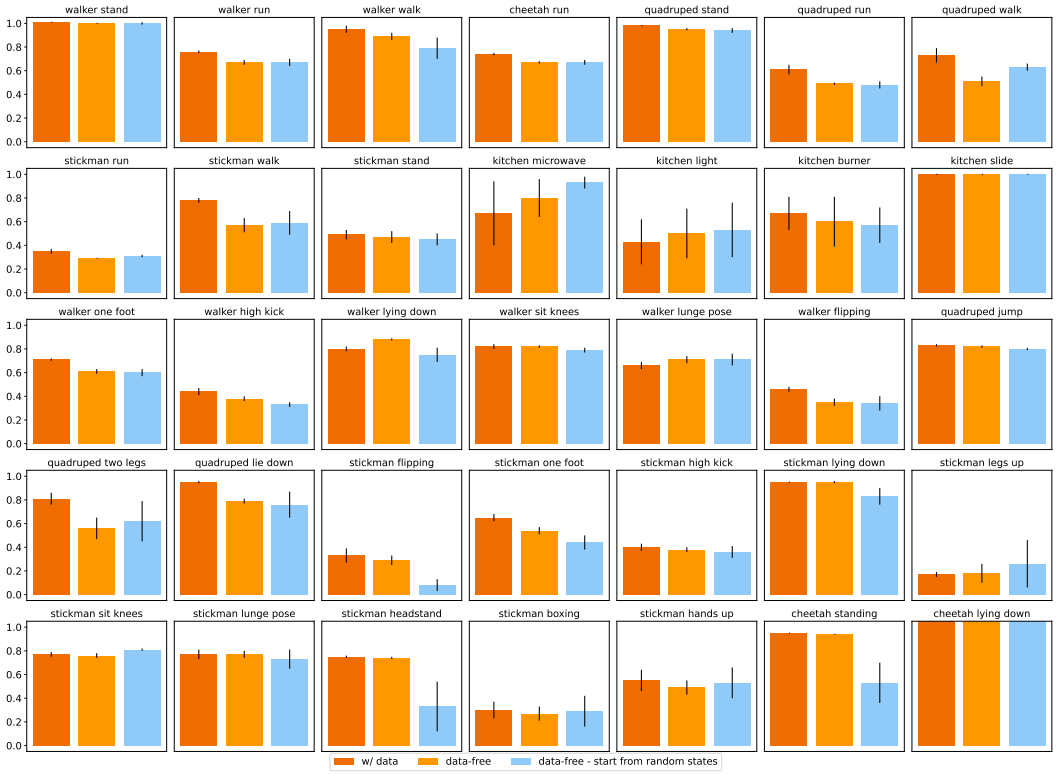


Figure 9: *Data-free RL detailed results*. Results averaged over 5 seeds.

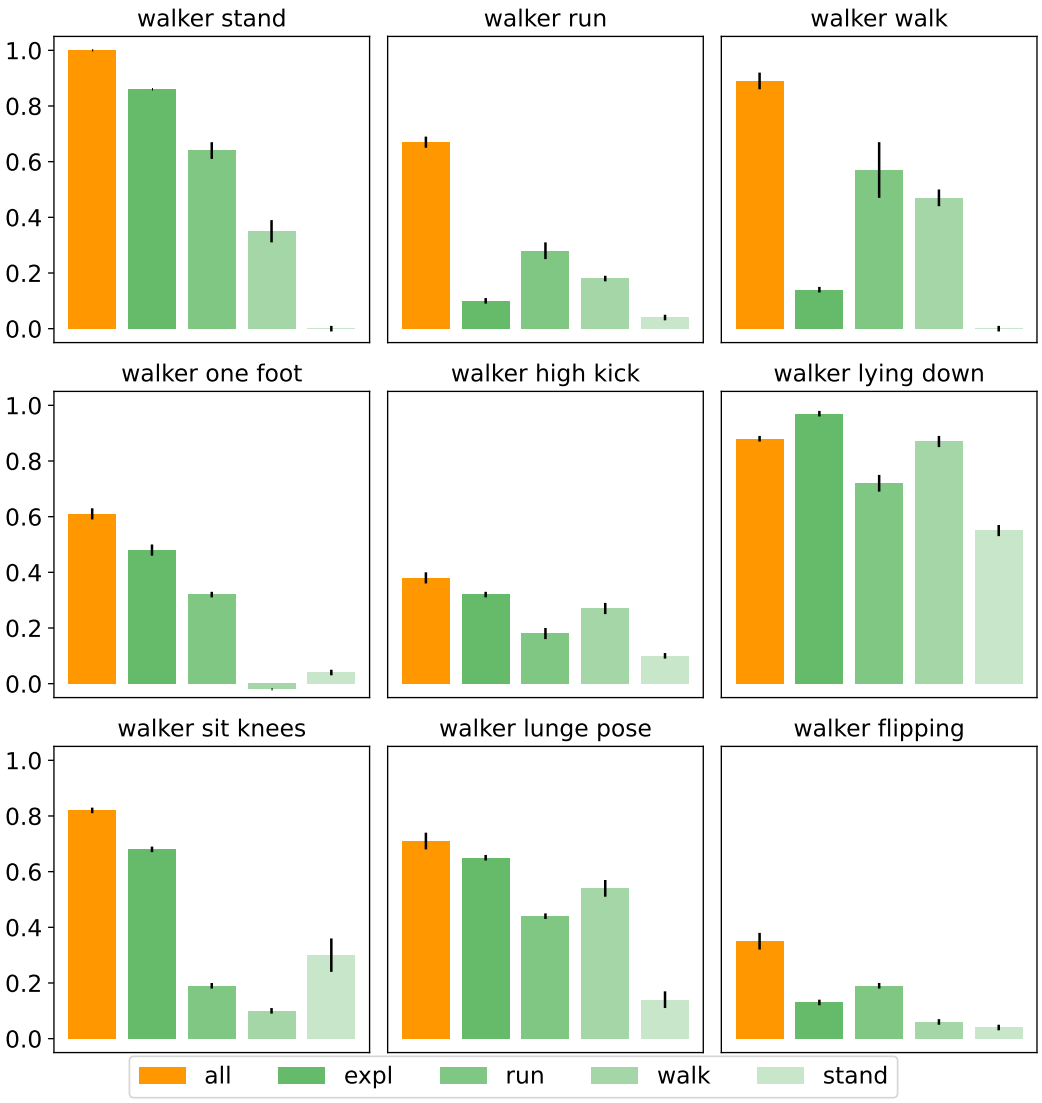


Figure 10: *Training data distribution detailed results*. Results averaged over 3 seeds.